

Amendments to the Specification

Please amend the paragraph on page 4, lines 8-17, as follows:

In one implementation of this embodiment, when it is certain (or maybe even just likely) that a relatively large amount of data being used or about to be used ~~use~~ will become garbage in a finite or predetermined period of time, the bottom of the window may stop moving, while the top of the window is allowed to expand to accommodate new objects. After the data becomes garbage and the young generation is garbage-collected and possibly compacted, the window may return to its normal size and behavior. Thus, as a result of temporarily sticking or holding the bottom of the window, the system avoids over-promotion and may reduce the frequency with which an older generation of memory needs to be garbage-collected.

Please amend the paragraph on page 4, lines 18-25, as follows:

In different embodiments of the invention, different events or knowledge may be cause for sticking the window bottom. For example, when an application, method or function is compiled or about to be compiled, the window bottom may be held in place until compilation is complete, particularly if the compilation is known to use or produce a large amount of temporarily data. As another example, when a graph or model is to be computed or generated using a large amount of temporary data, the window bottom may be stuck in place until the data are no longer needed.

Please amend the paragraph on page 7, lines 6-10, as follows:

In other embodiments of the invention, other events or knowledge may cause the first and second signals to be issued. In general, when a relatively ~~relative~~ large amount of data is to be stored for a finite period of time, the first signal is raised; the second signal follows when the data are no longer needed (i.e., they are garbage).

Please amend the paragraph on page 9, lines 1-5, as follows:

In operation 212, one end (e.g., the lower end) of the young window is fixed (i.e., not permitted to slide) and the garbage collector begins the temporary phase of memory

allocation. The garbage collector may receive information regarding the amount of data to be stored and/or the amount of time (or an absolute time) after which the data are expected or certain to become garbage.

Please amend the paragraph on page 10, lines 9-19, as follows:

The telephone may be configured with a JVM to interpret or execute various code (e.g., sounds, games, multimedia applications) that may generate temporary data. For example, a method or function of a program may be compiled by the JVM to speed its execution. The JVM may know that compilation will produce a significant amount of short-lived data. Thus, the JVM may instruct the garbage collection scheme to implement a temporary allocation phase of operation, and may signal for normal allocation to resume when compilation is complete. When temporary allocation is initiated, the young window may be garbage collected and adjusted to have its lower bound at the address where the allocation point was when the signal was received to begin temporary allocation.